Journal of Manufacturing Engineering, 2007, Vol.2, Issue.4

SELECTION OF OPTIMAL SEQUENCE OF MACHINING OPERATIONS IN CAPP

Dr.A.Gopala Krishna[1], E.Venu Gopal Goud[2] and C.Govinda Rajulu[3]

[1]Associate Professor, Department of Mechanical Engineering, J.N.T. U, College of Engineering, Kakinada – PIN 533 003, Andhra Pradesh, India

[2]Associate Professor, Department of Mechanical Engineering, G.P.R. Engineering College, Kurnool – PIN 518 003, Andhra Pradesh, India

[3]Associate Professor, Department of Mechanical Engineering, D.V.R. College of Engineering & Technology, Kashipur – PIN 502 285, Andhra Pradesh, India

ABSTRACT

Computer aided process planning (CAPP) is an important interface between computer aided design (CAD) and computer aided manufacturing (CAM). CAPP in the past is typically a knowledge-based approach, which is only capable of generating a feasible plan for a given part. The present work involves the application of a global search technique called Differential evolution for a quick identification of optimal or near optimal operation sequences. Minimization of sum of the total number of set-up changeovers and tool changeovers is taken as the objective function. Initially, the given part is represented as an assembly of form features, with details of geometric specifications, tolerance and surface finish requirements. To produce each of the form features, the required machining operations are selected. Next, feasibility constraints are considered among various machining operations. The proposed method then finds the optimal sequences within the minimum possible time.

Keywords: CAPP, Operations sequence, Differential Evolution

1. INTRODUCTION

Computer aided process planning (CAPP) is defined as an activity that translates part design specifications from an engineering drawing into manufacturing operation instructions required to convert a product from its initial stage to predetermined shape. There are two basic approaches to CAPP: Variant and Generative. Variant approach is based on Group technology concepts like Classification and coding systems to select generic process plan from the existing master process plans developed for each part family and edits to suit the requirement of the part. In the Generative approach, synthesizing the part data with the information from manufacturing databases and decision rules generates a process plan. In any CAPP system, selection of the operations sequence is an essential activity for manufacturing a part economically. Although there exists a large number of CAPP systems in the literature, only a few have taken into consideration, the optimization of operation sequencing and the alternate sequence of operations. To determine the optimal sequence, Integer programming [1], Branch and Bound method [2] and Dynamic programming [3] have been used. Consideration of applicable constraints made the formulation of problem very difficult and is NP-hard in nature. This limited the application of the conventional methods. Shunmugam *et al* [4] applied Genetic algorithms for generating the optimal sequence of manufacturing operations. The present work proposes the application of a recently emerged metaheuristic called Differential evolution (DE). In his earlier work [5], the author has applied DE for finding the optimal machining parameters of a complex machining problem and obtained better results than the existing methods.

2.IDENTIFICATION OF FORM FEATURES

The form features of a product can be classified into primary, secondary and C-axis features. Primary features (cylinder, cone, face) represent the basic shape of the part. The secondary features (threads, grooves, chamfer, slot) and C-axis features (radial hole, key hole) reside on the primary features and add details to the shape of the part.

Corresponding author: agopalakrishna@rediffmail.com

Depending on the type of form features, there may be a need to change the machining parameters, tool setups or machine tool.

3. CONSTRAINTS

A feasible sequence is the one, which does not violate any of the following feasibility constraints:

3.1 LOCATION CONSTRAINT

It is concerned with the examination of defined part features to determine which reference face is to be used to locate a particular feature. It refers to plane surfaces such as end or a face. The locating surface is to be machined prior to the reference surface. For example, for a rotational part, end face could be the reference feature.

3.2 PRE-CONDITION OR NONDESTRUCTIVE CONSTRAINT

This is taken into consideration so that the current machined features do not destroy the properties of features machined previously. For example, when thread and chamfer are two secondary features on a cylinder, chamfering must precede threading.

3.3 DATUM-HOLDING OR GEOMETRIC TOLERANCE CONSTRAINT

This refers to the datum requirements on features according to the geometric tolerance scheme. It results in the identification of features which must be machined in the same set-up, otherwise it would increase set-up time and cost.

3.4 ACCESSIBILITY CONSTRAINT

It establishes precedence constraint among the associated features, since a secondary feature is defined as the one, which resides on primary feature. It is not possible to machine the secondary feature until the primary feature is formed.

4.DIFFERENTIAL EVOLUTION - PROPOSED METHOD

Recently, evolutionary algorithms have received a lot of attention for solving a wide range of non-linear optimization problems. Evolutionary algorithms (EAs) mimic the metaphor of natural biological evolution, which adapt changing environments to find the optimum of a problem through evolving a population of candidate solutions. Among EAs, Genetic algorithms (GA) have got a lot of popularity in the recent past. Although several GA versions have been developed, they are not efficient when convergence speed is taken into consideration.

DE is an exceptionally simple, fast and robust evolutionary computation method proposed by Storn and Price [6] and is more likely to find the true optimum of a problem.

The following section brings out the major differences between GA and DE. In GA, all offspring are accepted and their parent strings are abandoned at the end of every generation regardless of their fitness values. This gives rise to a risk that a good parent string may be replaced with its deteriorated child string. Thus the improvement on the average performance of child population over parent population cannot be always guaranteed. This does not occur in DE, as the child string has to compete with its parent to get place in the succeeding generation. Second, in GA, parent strings with good fitness values are only be given chance to produce offspring without any consideration of the possibilities of generating better offspring by others. In DE, all solutions get the same chance of being selected as the parents with out dependence of their fitness values. These characteristics make DE perform better than GA.

Currently there are several versions of DE which are classified based on the notation *DE/x/y/z* where x specifies the vector to be mutated, y is the number of difference vectors considered for mutation of x, and z stands for the crossover scheme. x represents either 'rand' (randomly chosen vector) or 'best' (best vector of current population). For mutation, either single or two vector differences is used and accordingly y becomes either 1 or 2. z is either 'bin' (binomial) or 'exp' (exponential) depending upon the type of crossover scheme used.

Price and Storn [7] have suggested ten different working strategies in total and also some guidelines for applying them to a specific problem. The variant *DE/rand/1/bin* has been used in present work. The operators involved in the algorithm are explained in the following section.

4.1. MUTATION

In this phase, DE generates a mutant vector ($v_{i,g+1}$), by adding a weighted difference of two population vectors to a third vector using the following equation:

$$v_{i,g+1} = x_{r1,g} + F(x_{r2,g} - x_{r3,g})$$

where F>0 is a scaling factor, which controls the magnitude of the differential variation of ($x_{r2,g}$ - $x_{r3,g}$). The vectors x_{r1}, x_{r2}, and x_{r3} are to be randomly selected and to be different from the current vector. Unlike other EAs, where perturbation occurs in accordance with a random quantity, DE uses weighted difference between solution vectors (target vectors) to perturb the population at each generation as expressed in the above.

4.2 CROSSOVER

Crossover is introduced in the algorithm to control the amount of diversity of mutant vectors. Mutant vector and target vector are subjected to crossover to generate trial vector ($u_{i,g+1}$) based on the following equation:

$$u_{ji,g+1} = \begin{cases} v_{ji,g+1} & \text{if } rn_j \le C_r \\ x_{ji,g} & \text{otherwise} \end{cases}$$

$$\text{where } j = 1,2,---D$$

C_r Є [0,1] is the crossover constant which represents the probability of trial vector that inherits parameter values from the mutant vector and D represents the number of dimensions of a vector.

4.3 SELECTION

The trial vector produced by the crossover operator is compared with the target vector to determine the member for the next generation. If the trial vector produces a smaller function value it is copied to next generation otherwise target vector is passed into next generation.

$$\text{If } \quad f(u_{i,g+1}) \le f(x_{i,g}), \text{ set } x_{i,g+1} = u_{i,g+1}$$

$$\text{Otherwise } \quad x_{i,g+1} = x_{i,g}$$

One iteration constitutes the execution of all the above three operators in sequential order. The procedure continues until a stopping criterion is met. This criterion can be either to have the current best objective function value smaller than a specified value or the number of generations is equal to a predetermined maximum value.

4.4 CHOICE OF CONTROL PARAMETERS

Some guidelines are available in ref [8] to choose the control parameters *N*, *F*, and C_r . Normally, *N* is to be between 5 to 10 times the numbers of parameters in solution vector. *F* usually takes a value that ranges from 0.4 to 1.0. *F*=0.5 is a good initial choice and it can be increased if the population converges prematurely. On the other hand, a good value for Cr is 0.1; however to speed up convergence a greater value can be used. Although DE is used widely for continuous problems, yet it can also be used for solving NP-hard combinatorial problems. To solve the present combinatorial problem, the smallest position value (SPV) rule suggested by Tasgetiren et al. [9,10] is adopted. This is used to convert a continuous position vector into a permutation problem.

5.CASE STUDY

The part shown [11] in Figure 1 is taken as an example. Dimensional and geometrical tolerances along with other technical specifications are shown on the part drawing. The features involved in manufacturing the part are listed in Table 1. Each feature is coded with a numerical letter. Table 2 exhibits the constraints obtained for the part.

The objective is to determine the sequence of machining operations that corresponds to the minimum set-up change-overs and minimum tool change-overs. The following section illustrates the Differential Evolution (DE) algorithm as applied to the present problem.

Table 3 illustrates an example of the solution representation of target vector $x_{ij,g}$. According to the SPV rule, the smallest parameter value is $x_{ij,g}$ = 0.169, so the dimension *j*=3 is assigned to be the first feature P_{i1} = 3.The second smallest parameter value is 0.202, so the dimension *j* =10 is assigned to be the second feature and so on. In other words, dimensions are sorted according to the SPV rule, i.e., according to the parameter values $x_{ij,g}$ to construct the permutation $p_{ij,g}$. This enables the algorithm to work with the floating values. Total score for each sequence is calculated as weighted sum of individual scores using the following equation:

$$u = w_1 u_h + w_2 u_a$$

The assignments for w_1, w_2 depend on user preference. In the present work, w_1, w_2 are taken to be 3 and 2 respectively.

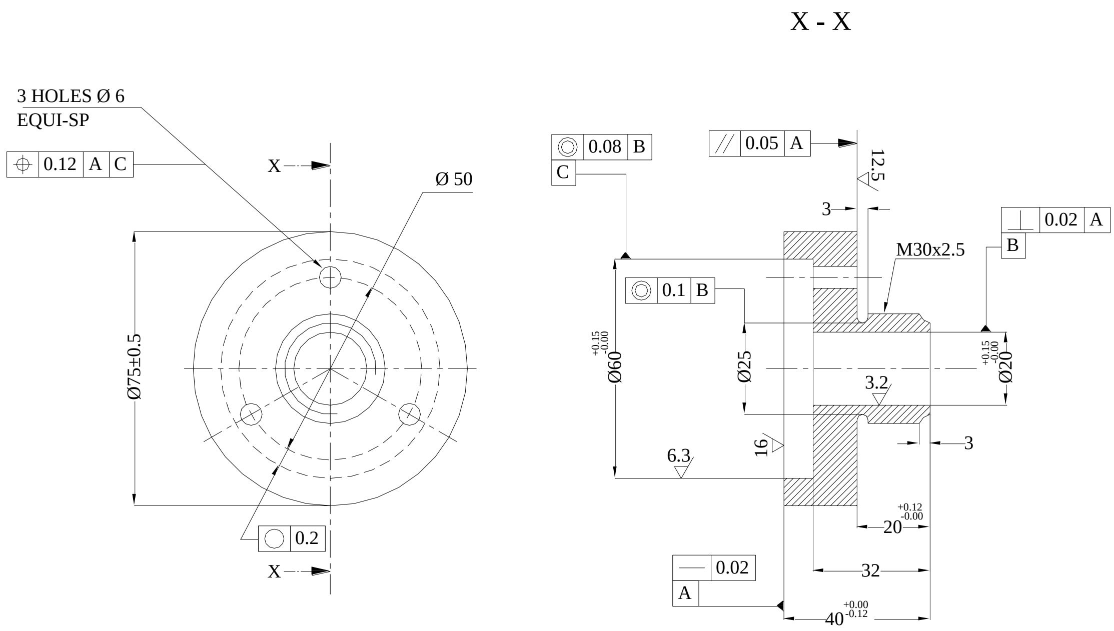

Fig.1.Flange

Table 1: Details of Features

Identification number	Name of Feature
0	End Face
1	End Face
2	External cylinder
3	Turn
4	Step Bore
5	Hole ϕ19.5
6	Chamfer
7	Thread
8	Cuts
9	Hole ϕ6
10	Step Face

Table 2: Constraints

Locating constraints	$0 \rightarrow 2$, $0 \rightarrow 5$, $0 \rightarrow 4$, $1 \rightarrow 3$, $1 \rightarrow 7$, $1 \rightarrow 6$, $1 \rightarrow 8$, $1 \rightarrow 9$, $1 \rightarrow 10$
Accessibility constraints	$0 \rightarrow 2$, $3 \rightarrow 10$, $2 \rightarrow 10$, $5 \rightarrow 4$, $3 \rightarrow 7$, $3 \rightarrow 8$, $3 \rightarrow 6$
Non-destruction constraints	$6 \rightarrow 7$, $8 \rightarrow 7$, $8 \rightarrow 10$
Geometric tolerance constraints	$5 \rightarrow 4$, $0 \rightarrow 5$, $0 \rightarrow 10$

Table 3: Solution representation of target vector

Dimension	$x_{ij,g}$	**Feature** ($p_{ij,g}$)
0	0.500	3
1	0.859	10
2	0.365	8
3	0.169	2
4	0.569	0
5	0.866	4
6	0.611	6
7	0.639	7
8	0.233	1
9	0.955	5
10	0.202	9

Table 4: Details of holding the features in the same setup

Set 1	0,2,5,4
Set2	3,8,6,7,10,9,1

Table 5: Adjacency templates

S.No	**Sequence of features**
1	$0 \rightarrow 2$
2	$1 \rightarrow 3 \rightarrow 10$

In the above equation, u_h represents number of changes in each sequence and

$u_a = \sum$ [number of out-of sequence features in adjacency-template (i)]

where i represents the template under construction.

Holding score: It is based on change of datum/reference features for holding the part for given sequence. Table 4 shows the features that can be machined in the same set-ups.

Adjacency score: It is based on the adjacency of features and is evaluated for each sequence based on the number of the elements not matching with the sequence given in each template. Table 5 lists the adjacency templates.

6. RESULTS AND DISCUSSION

The proposed technique is coded in JAVA and executed on a Pentium IV with 2.8 GHz processor. This problem has multiple solutions. The various optimal sequences obtained are 0-2-5-4-1-3-8-6-7-9-10,0-2-5-4-1-3-8-6-10-7-9, and 0-2-5-4-1-3-6-8-7-10-9 with the total cost of 5 units. DE has taken only 10s to converge to the optimal solution.

7. CONCLUSIONS

For a computer-aided process planning system (CAPP) to handle a part comprising a large number of interacting features, an efficient algorithm is needed for exploring and reducing the size of the search space of valid operation sequences. This work presents an application of a recently developed global optimization technique, Differential Evolution algorithm (DE), for finding the optimal operation sequences. Although several conventional optimization techniques are available to solve the NP-hard problems in the literature, their application is quite limited because of the possibility of getting trapped at local-optimal points and because of lack of robustness. The proposed method overcomes the above drawbacks. DE has simple structure and is powerful and robust algorithm for solving the NP-hard problems. DE is more likely to find the function's true optimum than other methods. Similar to Genetic algorithms and Simulated annealing, DE is a completely generalized method, as it has no restrictive assumptions about the objective function, parameter set or constraint set.

8. REFERENCES

1. Lin,C J., and Wang, H P., 1993, "Optimal operation planning and sequencing; minimization of tool changeovers", Int J Prod Res 31(2): 311-324
2. Koulamas C., 1993, "Operation sequencing and machining economics", Int J Prod Res 31(4): 957-975
3. Halevi, G., and Weill, R., 1988, "Development of flexible optimum process planning procedures", CIRP 29(1): 313-317
4. Shunmugam, M, S., Mahesh Puneti., Bhaskar Reddy, S V., 2002, "A method of preliminary planning for rotational components with C -axis features using genetic algorithms",48,199-217.
5. Gopala Krishna, A., 2007, "Optimisation of surface grinding operations using a differential evolution approach", Jrl. of Materials Processing Technology, corrected proof available online.
6. Storn, R., Price, K., 1997,"Differential evolution–a simple evolution strategy for fast optimization", Dr. Dobb's J., 15 18-24.
7. Price K, and Storn R, Differential evolution home page (web site of Price and Storn) http://www.icsi.berkely.edu/~storn /code.html as at 2001.
8. Price, K., 1999. An introduction to differential evolution, In: Corne D, Dorigo M, Glover (Eds.), New Ideas in Optimization, McGraw Hill, International (UK), pp. 79–108.
9. Tasgetiren, M,F., Sevkli, M., Liang, Y-C., Gencyilmaz, G., "Particle swarm optimization algorithm for single-machine total weighted tardiness problem", Congress on Evolutionary Computation",CEC2004, Portland, Oregan, USA
10. Tasgetiren, M,F., Liang, Y-C., Sevkli, M., Gencyilmaz,G., "Particle swarm optimization algorithm for permutation flowshop sequencing problem", Fourth International Workshop on Ant Algorithms and Swarm Intelligence, ANTS2004, Brussel, Belgium.
11. Narayana, K. L., Kannaiah, P., Venkata Reddy,K., Reprint, 2001, Production Drawing, New Age International Publishres, New Delhi.